

**О.Г. ВАСИЛЬЧЕНКОВ**, канд. тех. наук, доц. каф. АУТС НТУ «ХПИ»,  
**А.С. МАТВИЕНКО**, студентка НТУ «ХПИ» (г. Харьков)

## **СОВРЕМЕННЫЕ МЕТОДЫ ВИЗУАЛИЗАЦИИ РЕАЛИСТИЧНЫХ ТРЕХМЕРНЫХ ЛАНДШАФТОВ В ТРЕНАЖЕРНЫХ СИСТЕМАХ ИМИТАЦИИ ДВИЖЕНИЯ**

У статті описані основні сучасні методи візуалізації реалістичних тривимірних ландшафтів у тренажерних системах імітації руху, їхні достоїнства й недоліки. Показано напрямок подальших досліджень для вдосконалення існуючих методів.

In papers the basic modern methods of visualisation of realistic three-dimensional landscapes in training systems of imitation of movement, their merits and demerits are reviewed. The direction of the further researches for improvement of existing methods is shown.

В тренажерных системах имитации визуальной обстановки можно выделить две составляющих – создание самой сцены и визуализация ее в реальном режиме времени во время проведения тренировки. Для подготовки объектов сцены часто используют такие промышленные программные продукты, как *3D Studio* и *3D MAX*. Однако эти системы не позволяют визуализировать сцену в реальном режиме времени. Для этих целей приходится создавать отдельные программные средства.

Рендеринг ландшафтов является сложной проблемой при программировании реалистичной графики. В настоящий момент мы находимся в интересной точке развития технологий рендеринга (т.е. "оцифровки" данных, по которым строится ландшафт), поскольку увеличившаяся пропускная способность современных видеокарт в совокупности с опубликованными исследованиями на тему алгоритмов реального времени *LOD meshing* дают возможность современным графическим движкам отрисовывать сильно детализированные ландшафты. Тем не менее большинство используемых технологий предлагают компромисс между размером ландшафта и его детализированностью. Рассмотрим три основных метода создания ландшафтов:

### **1. Метод создания ландшафта заданием экстремумов.**

Алгоритм создания ландшафта с помощью задания экстремумов:

- Создать двумерный массив вершин.
- Задать экстремумы. Экстремумы задаются над полигонной сеткой ландшафта. Там, где будет располагаться экстремум появятся горы или впадины.

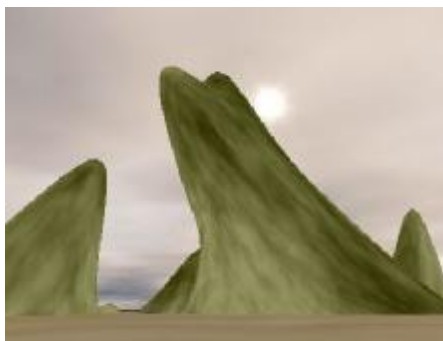


Рис. 1. Ландшафт созданный с помощью задания экстремумов

- Задать "радиус распространения" каждого экстремума. Т.е. любая гора в ширину достигает определенного значения, это значение –максимальный диаметр горы.

- Вычислить расстояние от каждой вершины до каждого экстремума. Чем меньше расстояние, тем выше поднимается точка (при расстоянии равном 0 высота точки будет равна высоте экстремума). Точки, которые находятся на расстоянии большем чем "радиус распространения", не входят в образование горы и остаются на своем месте.

- При задании экстремумов каждому из них задается значение "сдвига" по  $x, z$ . Иначе говоря задаются два параметра  $dX, dZ$ . Когда будет вычислено на какую высоту ( $dY$ ) поднять точку, тогда значение высоты умножается на коэффициенты сдвига по  $x, z$  ( $dX, dZ$ ) и получившееся значение складывается с  $X, Z$  координатами точки.

В результате получается массив вершин, который задаёт полигонную сетку.

## 2. Создание ландшафта с помощью фракталов.

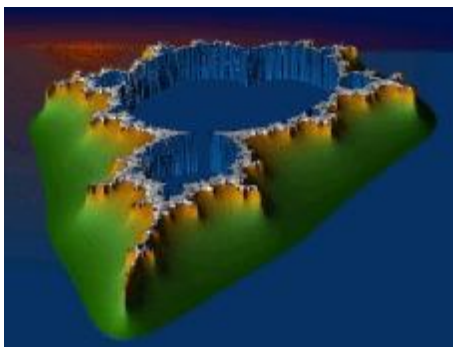


Рис. 2 – Ландшафт созданный с помощью фракталов

В современной компьютерной графике широко применяются фракталы. Фрактальная геометрия незаменима при генерации искусственных облаков, морей, горных ландшафтов. Фракталам подвластны самые невероятные формы, ведь компьютерное искусство фрактальной геометрии не знает границ.

Рассмотрим метод построения фрактальной кривой с помощью компьютера и его математическое обоснование на примере кривой Леви. Базовым элементом для кривой Леви является отрезок  $AB$ . Первый шаг включает в себя построение равнобедренного прямоугольного треугольника на отрезке  $AB$  как на гипотенузе. При этом сама гипотенуза в дальнейшем построении не участвует и исключается из кривой. Таким образом, первое поколение кривой представляет собой ломаную  $ACB$  (рис.3).

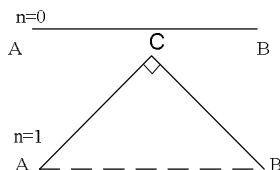


Рис. 3

На втором шаге с каждым из отрезков  $AC$  и  $CB$  проделываем ту же операцию, что и с отрезком  $AB$  на первом шаге. Последующие поколения кривой строятся по тому же алгоритму. Следовательно, задача сводится к определению координаты точки  $C(x,y)$  по известным координатам точек  $A$  и  $B$ .

Вспомогательная задача:

Вектор  $\vec{a}$  задан своими координатами  $\{x_a; y_a\}$ . Известно, что вектор  $\vec{a}$  перпендикулярен вектору  $\vec{b}$  и  $|\vec{a}| = |\vec{b}|$ . Найти координаты вектора  $\vec{b}$ .

*Решение:*

Пусть вектор имеет координаты  $\{x_b; y_b\}$ . Тогда из условия следует, что

$$\vec{a}\vec{b} = x_a x_b + y_a y_b = 0;$$

$$|\vec{a}| = |\vec{b}|;$$

$$x_a^2 + y_a^2 = x_b^2 + y_b^2.$$

Получим систему уравнений с двумя неизвестными:

$$\begin{cases} x_a x_b = -y_a y_b \\ x_a^2 + y_a^2 = x_b^2 + y_b^2 \end{cases};$$

$$\left\{ \begin{array}{l} x_b = \frac{-y_a y_b}{x_a} \\ x_a^2 + y_a^2 = x_b^2 + y_b^2 \end{array} \right. ; \quad \left\{ \begin{array}{l} x_b = \frac{-y_a y_b}{x_a} \\ x_b^2 + \left( \frac{-y_a y_b}{x_a} \right)^2 = x_a^2 + y_a^2 \end{array} \right. .$$

Рассмотрим второе уравнение системы:

$$x_b^2 + \frac{-y_a^2 y_b^2}{x_a^2} = x_a^2 + y_a^2, \quad x_a^2 y_b^2 + y_a^2 y_b^2 = x_a^4 + x_a^2 y_a^2,$$

$$y_b^2 (x_a^2 + y_a^2) = x_a^2 (x_a^2 + y_a^2),$$

$$y_b^2 = x_a^2,$$

$$\begin{cases} y_b = x_a \\ y_b = -x_a \end{cases}.$$

Возвращаясь к системе, имеем

$$\begin{cases} y_b = x_a \\ x_b = -\frac{y_a x_a}{x_a} \end{cases}; \quad \begin{cases} y_b = x_a \\ x_b = -y_a \end{cases}$$

или

$$\begin{cases} y_b = -x_a \\ x_b = y_a \end{cases}$$

Таким образом, если  $\vec{a}\{x_a; y_a\}$ , и вектор  $\vec{a}$  перпендикулярный вектору  $\vec{b}$ ,  $|\vec{a}| = |\vec{b}|$ , то  $\vec{b}\{-y_a; x_a\}$  или  $\vec{b}\{y_a; -x_a\}$

Вернемся к построению кривой Леви.

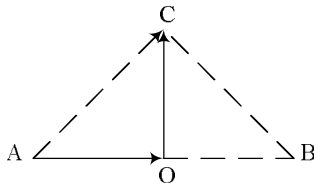


Рис. 4

По свойству прямоугольного равнобедренного треугольника (рис. 4) точки  $A, B, C$  равноудалены от точки  $O$  – середины гипотенузы, отрезок  $OC$  перпендикулярен отрезку  $AO$ ,  $OC = AO$ . Пусть  $A(x_0, y_0); B(x_1, y_1)$ , тогда точка

$O$  как середина отрезка  $AB$  будет иметь координаты  $O\left(\frac{x_0 + x_1}{2}; \frac{y_0 + y_1}{2}\right)$ .

Применяя вспомогательную задачу, имеем  $\vec{a} = \vec{AO}$ ,  $\vec{b} = \vec{OC}$ . Т.к.  $\vec{AO}\left(\frac{x_1 - x_0}{2}; \frac{y_1 - y_0}{2}\right)$ , то  $\vec{OC}\left(-\frac{y_1 - y_0}{2}; \frac{x_1 - x_0}{2}\right)$  или  $\vec{OC}\left(\frac{y_1 - y_0}{2}; -\frac{x_1 - x_0}{2}\right)$ . Для построения кривой Леви необходим вектор, направленный влево от  $\vec{AO}$ , т.е.  $\vec{OC}\left(-\frac{y_1 - y_0}{2}; \frac{x_1 - x_0}{2}\right)$ . Если  $O\left(\frac{x_0 + x_1}{2}; \frac{y_0 + y_1}{2}\right)$  и  $C(x; y)$ , то

$$x = \frac{x_0 + x_1}{2} - \frac{y_1 - y_0}{2}, \quad y = \frac{y_0 + y_1}{2} + \frac{x_1 - x_0}{2}.$$

$$\text{Значит, } C\left(\frac{x_0 + x_1}{2} - \frac{y_1 - y_0}{2}; \frac{y_0 + y_1}{2} + \frac{x_1 - x_0}{2}\right).$$

Поскольку построение фрактальной кривой состоит в генерации поколений по одному и тому же алгоритму, то координаты искоемых точек будут определяться по рекуррентной формуле:

$$x = \frac{x_i + x_{i-1}}{2} - \frac{y_i - y_{i-1}}{2}, \quad y = \frac{y_i + y_{i-1}}{2} + \frac{x_i - x_{i-1}}{2}.$$

Здесь  $(i-1)$ -я точка является аналогом точки  $A$ , а  $i$ -я – аналогом точки  $B$ .

На основании проведенных вычислений строится программа вычисления координат точек-вершин фрактальной кривой.

При ее составлении необходимо учесть следующий математический факт, определяющий количество строящихся точек на каждом шаге.

### 3. Метод ROAM

Рассмотрим алгоритм меша с оптимальной подгонкой в реальном времени (*Real-Time Optimal Adapting Mesh*), больше известный как *ROAM*, (впервые он был представлен в работе Марка Дюшено (*Mark Duchaineau*) как алгоритм, созданный для упрощения визуализации больших участков ландшафта.

*ROAM* основан на одном из свойств равнобедренного прямоугольного треугольника. Как видно из рис. 5, такой треугольник можно поделить на два равных равнобедренных прямоугольных треугольника, разбив исходную фигуру по линии, соединяющей угол при вершине между равными сторонами с центром основания треугольника. Процесс деления можно продолжать бесконечно, всякий раз удваивая число треугольников. Коль скоро каждый из треугольников является потенциальным родителем двух меньших фигур, то формируется иерархия типа «бинарное дерево». *ROAM* – это, в основе своей, метод управления тем, какие треугольники делятся пополам, а какие сливаются и совпадают с родителем. Это позволяет увеличивать и уменьшать степень детализации ландшафта на уровне треугольников.



Рис. 5 – Прямоугольный треугольник

Каждый раз при разбиении треугольника на середине его основания (рис. 5) строится новая вершина. Её положение можно найти интерполяцией как среднее между конечными точками основания. Однако свое значение по оси  $z$  новая вершина наследует от меша высот, так поднимая или опуская ее, чтобы добиться согласия с базовыми ландшафтными данными.

**Выводы:** Приведенные выше методы генерации ландшафтов являются наиболее распространенными в настоящее время для построения систем визуализации тренажеров транспортных средств. Достоинствами методов экстремумов и фрактального являются простота программной реализации алгоритмов генерации, быстрая генерация ландшафтов, возможность изменений ландшафтов в реальном масштабе времени. С другой стороны зачастую ландшафты полученные такими методами далеки от реальных, что не делает их использование неприемлемым для целей обучения. Таким образом, *ROAM* метод является наиболее перспективным на сегодняшний день. Недостатками этого метода являются сложная реализация, низкая детализация ландшафтного объекта на больших расстояниях, что, в некоторых случаях, заметно визуально. Также проблемой метода является невозможность реализации больших перепадов высоты на небольшой площади (обрывы, отвесные скалы и т.д.).

Таким образом, разработка модифицированного метода *ROAM*, который уменьшает перечисленные недостатки, является перспективным направлением дальнейших исследований.

**Список литературы:** 1. Васильев В.Е., Морозов А.В. Компьютерная графика: Учебное пособие. – СПб.: СЗТУ, 2005. – 101с. 2. Грег Снук. Создание 3D- ландшафтов в реальном времени с использованием C++ и DirectX 9/ пер.с англ. – М.: КУДИЦ-ОБРАЗ, 2007. 3. Tomas Moller, Eric Haines «Real-TimeRendering» 2-е изд. Natick, Mass.: A K Peters, Ltd., 2002

Поступила в редколлегию 13.12. 2008 г.